

FEM-3

有限元大作业答辩

组长：黄云帆

组员：管唯宇 陈一彤 邓博元 杨正宇 卢昊

2017.12.27



目录

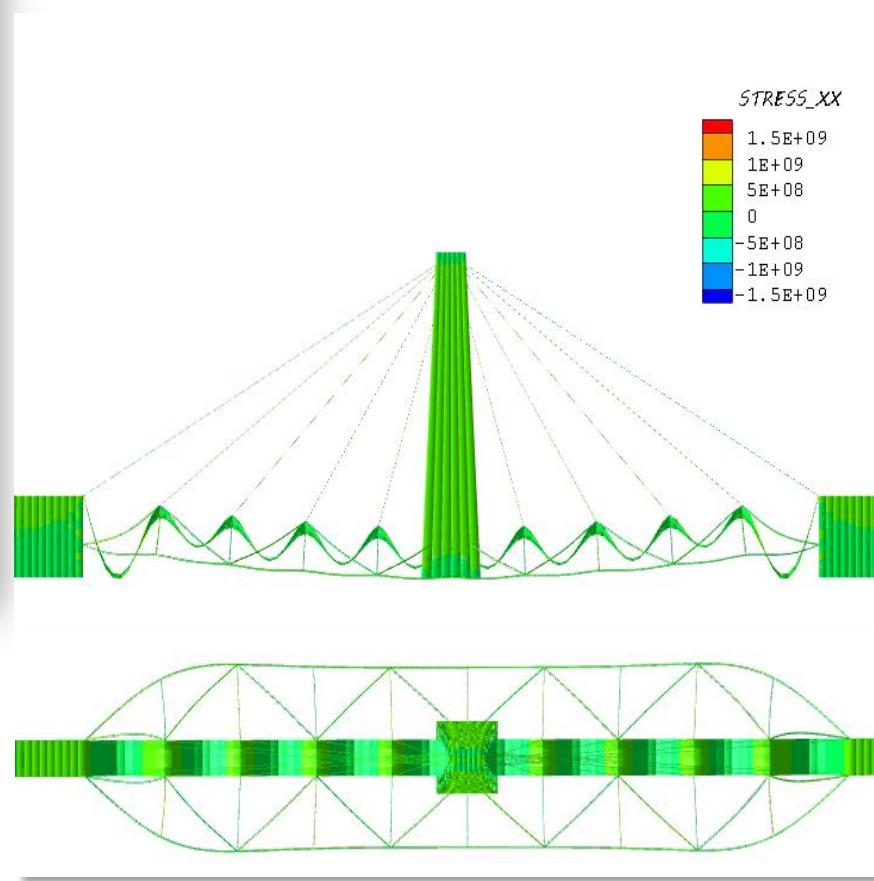
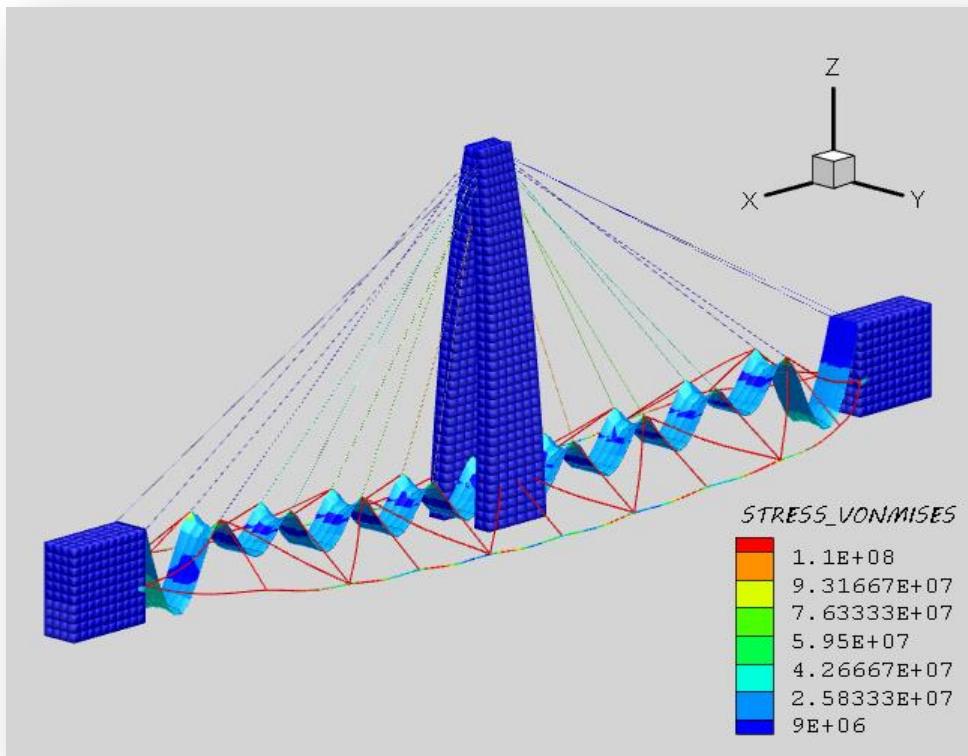
- 1 成果汇报
- 2 功能概述
- 3 单元构造
- 4 高级功能
- 5 致谢



成果汇报

1

成果汇报





1 成果汇报

经过一个月的努力，经历了 9 次每周五晚的常设例会以及若干次不定期讨论，在小组成员的紧密配合下，本组顺利地完成了大作业的任务。下面将以 STAP++ 程序为主进行展示。

算例的具体描述与测试结果如下^[1]：

算例	<i>NumNode</i>	<i>NumEle</i>	<i>S4R</i>	<i>C3D8R</i>	<i>B31</i>	<i>T3D2</i>	<i>t_{min}/s</i>	<i>RAM_{min}/M</i>
1	4.16E3	2.88E3	4E2	1.76E3	7.04E2	20	0.27	\
2	3.72E4	3.04E4	2.5E3	2.65E4	1.35E3	20	2.35	\
3	\	2.33E5	1E4	2.2E5	2.7E3	20	25.0 _{6244M}	1385 _{132.63s}
4	1.91E6	1.81E6	4E4	1.76E6	5.42E3	20	3131 _{12C}	\

¹环境：CPU: intel xeon cpu e5-2620 v4, 2.10Hz, 16核；内存: 64G；操作系统: win10；MSVC 编译, O_x 优化。



功能概述

前后处理 单元构造 求解器 模态分析

2 功能概述

- 前后处理
 - 前：Python 开发；DOF 自动适配
 - 后：Tecplot 作图；基于 8H 的 SPR 功能
- 单元构造
 - 基本单元：杆单元^{Bar}, 六面体实体单元^{8H}, 欧拉梁^{Beam}, 平板壳^{Shell}.
 - 其他单元：平面单元^{3T, 4Q, 9Q}, 纯弯板^{Plate}, 铁木辛柯梁^{EB, SR}, 截锥壳^{Frustum*}
 - 扩展单元：过渡单元^{5Q}, 无限单元^{InfEle}
- 求解器
 - LDLT 求解器
 - PARDISO® 稀疏求解器
- 模态分析



单元构造

实体单元 (杆元/平面/立体)

构造单元 (梁板壳)

扩展单元 (过渡/无限)

3

单元构造：实体单元-Bar, 3T, 4Q, 9Q, 8H

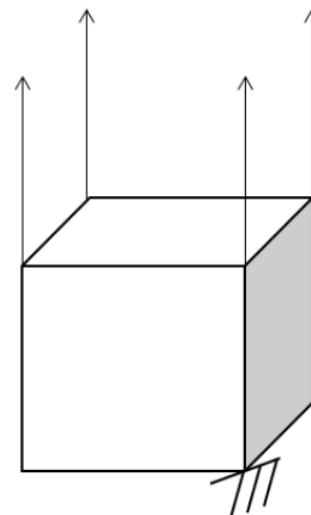
$$N_i = \frac{1}{8}(1 + \psi_i\psi)(1 + \eta_i\eta)(1 + \zeta_i\zeta)$$

$$i = 1, 2, \dots, 8$$

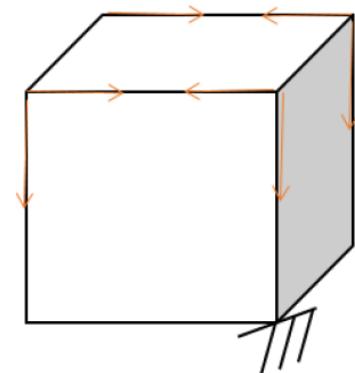
$$B_i = \begin{pmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{pmatrix}$$

$$B = [B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8]$$

$$K^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 B^T DB |J| d\psi d\eta d\zeta$$



Single axis stretch



Pure shearing

3

单元构造：实体单元-Bar, 3T, 4Q, 9Q, 8H

- 公式利用 mathematica / MATLAB 生成

```

In[1]:= a =  $\left\{ \frac{1}{2} \xi (\xi - 1), 1 - \xi^2, \frac{1}{2} \xi (\xi + 1) \right\};$ 
b =  $\left\{ \frac{1}{2} \eta (\eta - 1), 1 - \eta^2, \frac{1}{2} \eta (\eta + 1) \right\};$ 
indexs = {{1, 1}, {3, 1}, {3, 3}, {1, 3}, {2, 1}, {3, 2}, {2, 3}, {1, 2}, {2, 2}};
NN = Table[a[[item[[1]]]] * b[[item[[2]]]], {item, indexs}];
 $\quad$  |表格
x = {-1, 1, 1, -1, 0, 1, 0, -1, 0};
y = {-1, -1, 1, 1, -1, 0, 1, 0, 0};
f[i_] := NN /. { $\xi \rightarrow x[[i]]$ ,  $\eta \rightarrow y[[i]]$ }
Table[f[i], {i, 9}] == IdentityMatrix[9]
 $\quad$  |表格 |单位矩阵
GetLine1[n_] := Flatten[Table[{D[n[[i]],  $\xi$ ], 0}, {i, 9}]];
 $\quad$  |压平 |表格 |偏导
GetLine2[n_] := Flatten[Table[{0, D[n[[i]],  $\eta$ ]}, {i, 9}]];
 $\quad$  |压平 |表格 |偏导
GetLine3[n_] := Flatten[Table[{D[n[[i]],  $\eta$ ], D[n[[i]],  $\xi$ ]}, {i, 9}]];
 $\quad$  |压平 |表格 |偏导 |偏导
GetB[n_] := {GetLine1[n], GetLine2[n], GetLine3[n]};
GN = Transpose[Simplify@Table[{D[NN[[i]],  $\xi$ ], D[NN[[i]],  $\eta$ ]}, {i, 9}]];
 $\quad$  |转置 |化简 |表格 |偏导 |偏导
Out[8]= True

```

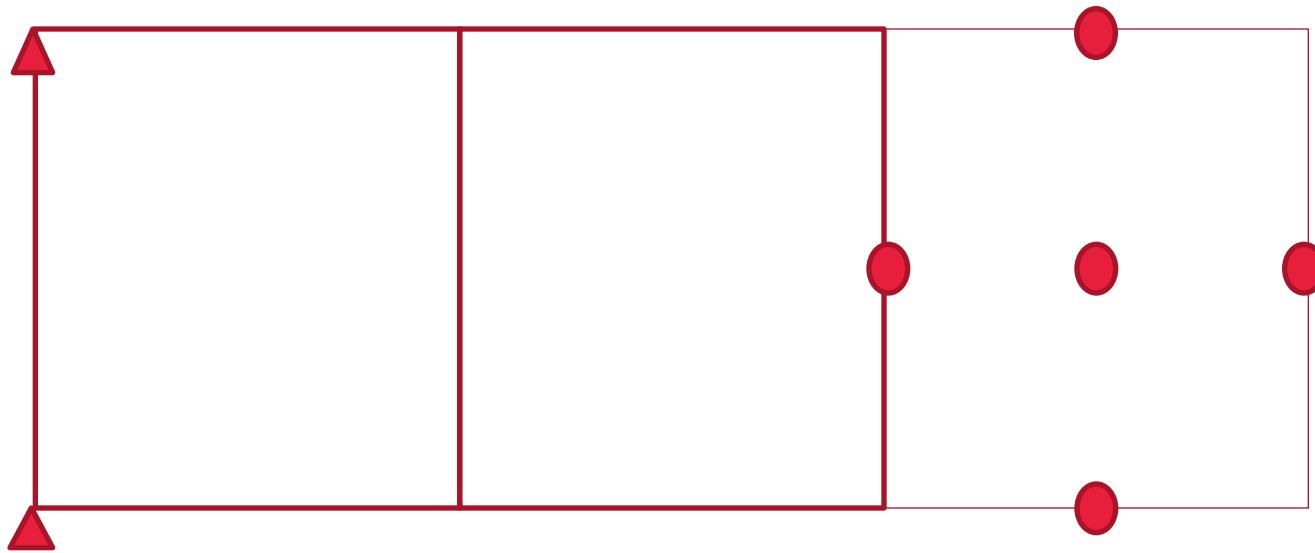
```

%% % Calculate invJe and detJ
% Construct symbolic J
syms J Je Jp0q;
J = sym(zeros(9, 1));
J(1) = Jp0q;
J(2:9) = sym('Jp%dq', [1, 8]);
Je = sym(zeros(3, 3));
num = 0;
for j = 1:3
    for i = 1:3
        num = num + 1;
        Je(i, j) = J(num, 1);
    end
end
% Calculate invJe and detJ
invJe = inv(Je);
detJ = det(Je);
% Transform invJe into a vector
num = 0;
invJ = sym(zeros(9, 1));
for j = 1:3
    for i = 1:3 ...
end
invL = ccode(invJ(:, 1))

```

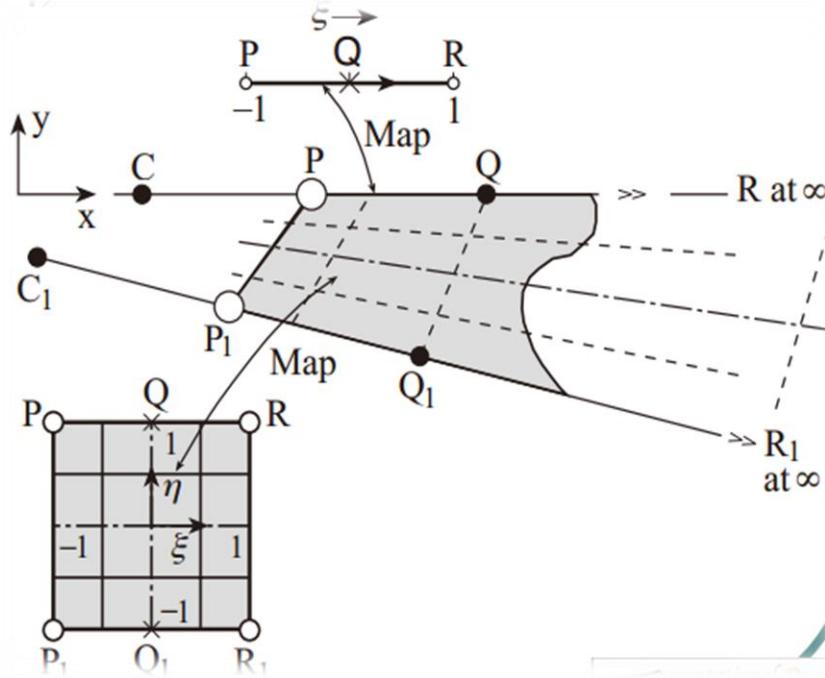
3

单元构造：扩展单元-5Q (4Q-9Q)



3

单元构造：扩展单元-无限单元



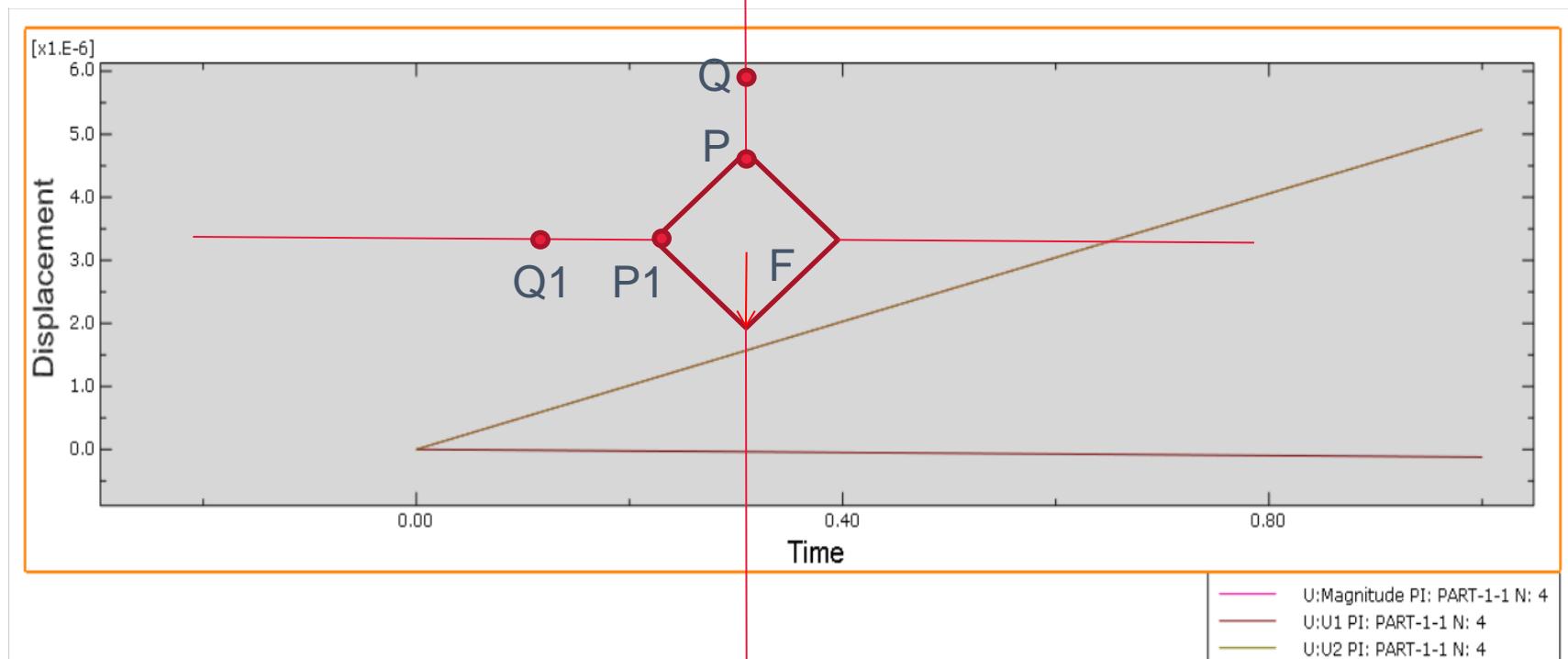
$$x = N_1(\eta)N_P(\xi)x_P + N_1(\eta)N_Q(\xi)x_Q + N_1(\eta)N_P(\xi)x_{P_1} + N_1(\eta)N_Q(\xi)x_{Q_1}$$

$$N_1 = \frac{1+\eta}{2}, N_2 = \frac{1-\eta}{2}$$

$$N_P = \frac{-2\xi}{1-\xi}, N_Q = \frac{1+\xi}{1-\xi}$$

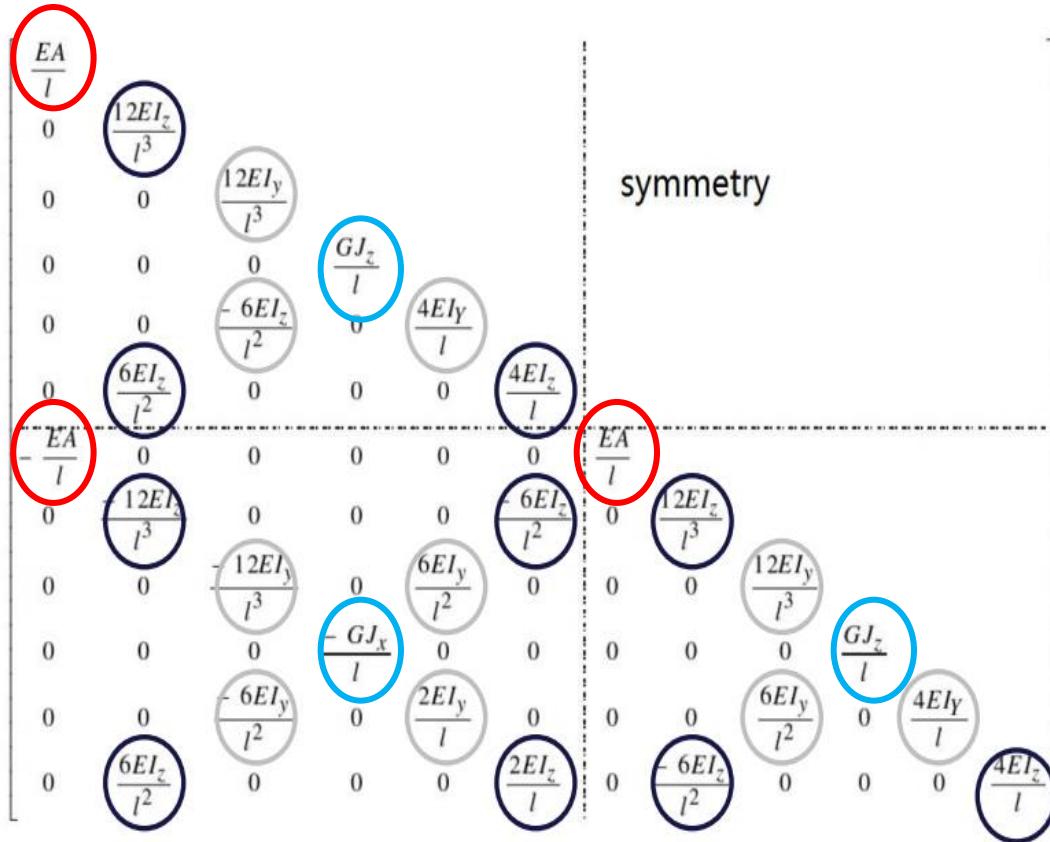
3

单元构造：扩展单元-无限单元



3

单元构造：构造单元-Euler 梁



关键：刚体模态约束

四个刚体模态

DOF自动适配功能

3

单元构造：构造单元-Timoshenko 梁

- 单元刚度阵的组装 (EBMOD; SRINT)

$$K = \begin{pmatrix} \text{Tens} & 0 & 0 & 0 & 0 & -\text{Tens} & 0 & 0 & 0 & 0 \\ \text{Bendz1} & 0 & 0 & 0 & \text{Bendz2} & 0 & -\text{Bendz1} & 0 & 0 & \text{Bendz2} \\ & \text{Bendy1} & 0 & -\text{Bendy2} & 0 & 0 & -\text{Bendy1} & 0 & -\text{Bendy2} & 0 \\ & & \text{Tors} & 0 & 0 & 0 & 0 & -\text{Tors} & 0 & 0 \\ & & & \text{Bendy3} & 0 & 0 & \text{Bendy2} & 0 & \text{Bendy4} & 0 \\ & & & & \text{Bendz3} & 0 & -\text{Bendz2} & 0 & 0 & \text{Bendz4} \\ & & & & & \text{Tens} & 0 & 0 & 0 & 0 \\ & & & & & & \text{Bendz1} & 0 & 0 & -\text{Bendz2} \\ & & & & & & & \text{Bendy1} & 0 & \text{Bendy2} \\ & & [symmetry] & & & & & & \text{Tors} & 0 \\ & & & & & & & & & 0 \\ & & & & & & & & \text{Bendy3} & 0 \\ & & & & & & & & & \text{Bendz3} \end{pmatrix}. \quad (2.8)$$

- $\text{Tens} = \frac{EA}{l}$, $\text{Tors} = \frac{GJ}{l}$;
- $\text{Bendy1} = \frac{GA}{kl}$, $\text{Bendy2} = \frac{GA}{2k}$, $\text{Bendy3} = \frac{GA}{4k}$ (or $\frac{GA}{3k}$) + $\frac{EI_y}{l}$, $\text{Bendy4} = \frac{GA}{4k}$ (or $\frac{GA}{6k}$) - $\frac{EI_y}{l}$;
- $\text{Bendz1} = \frac{GA}{kl}$, $\text{Bendz2} = \frac{GA}{2k}$, $\text{Bendz3} = \frac{GA}{4k}$ (or $\frac{GA}{3k}$) + $\frac{EI_z}{l}$, $\text{Bendz4} = \frac{GA}{4k}$ (or $\frac{GA}{6k}$) - $\frac{EI_z}{l}$.

其中, $b_z := \frac{12EI_z k}{GA l^2}$ 为剪切附加项, 对于矩形截面梁有 $k = \frac{6}{5}$

3

单元构造：构造单元-Timoshenko 梁

- 算例验证

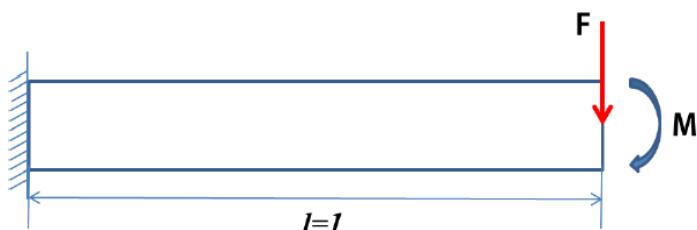


Figure 1: Example

$$\begin{cases} \frac{dM_z}{dx} + F_y = 0, \\ \frac{dF_y}{dx} + \bar{q} = 0. \\ M_z \equiv EI_z \kappa_z = EI_z \frac{d\theta_z}{dx}, \\ F_y \equiv \frac{GA}{k} \gamma_{xy} = \frac{GA}{k} \left(\frac{dv}{dx} - \theta_z \right). \end{cases}$$

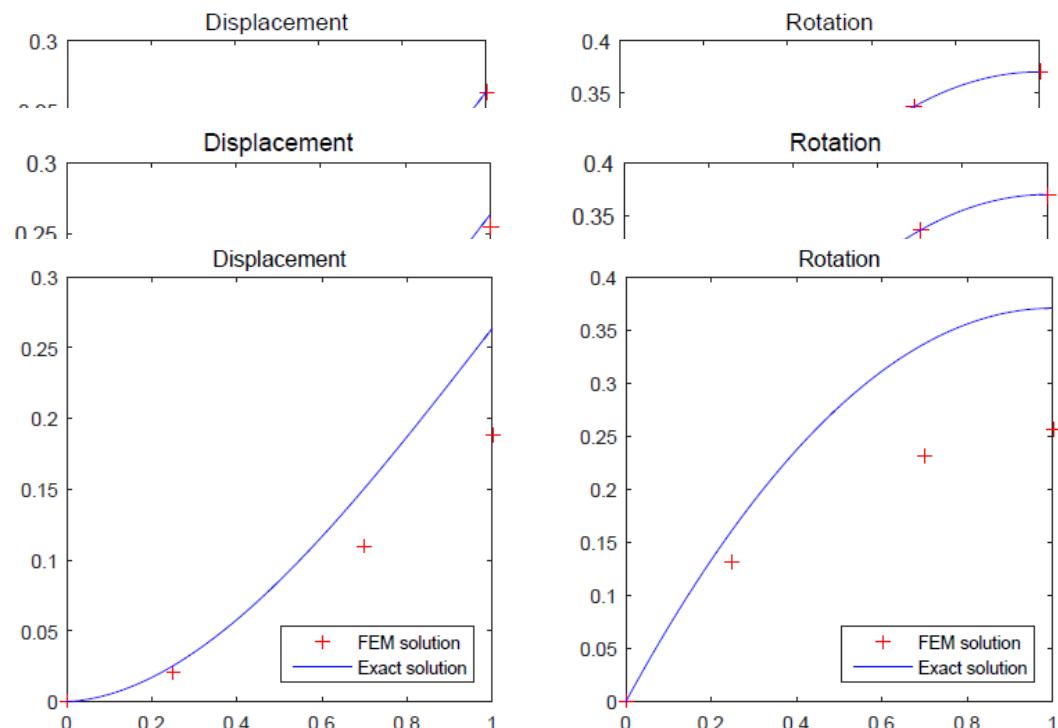


Figure 6: Concentrated F_y WITHOUT SRINT

3

单元构造：构造单元-矩形板

- 采用单元: 4节点12自由度矩形板
- 实际占用自由度数: 24自由度
- 输入参数: $E \gg v \gg h$

$$\mathbf{N}^T = \left\{ \begin{array}{l} -\frac{1}{8}(s-1)(t-1)(s^2+s+t^2+t-2) \\ -\frac{1}{8}b(s-1)(t-1)^2(t+1) \\ \frac{1}{8}a(s-1)^2(s+1)(t-1) \\ \frac{1}{8}(s+1)(t-1)(s^2-s+t^2+t-2) \\ \frac{1}{8}b(s+1)(t-1)^2(t+1) \\ \frac{1}{8}a(s+1)^2(s-1)(t-1) \\ -\frac{1}{8}(s+1)(t+1)(s^2-s+t^2-t-2) \\ \frac{1}{8}b(s+1)(t+1)^2(t-1) \\ -\frac{1}{8}a(s+1)^2(s-1)(t+1) \\ \frac{1}{8}(s-1)(t+1)(s^2+s+t^2-t-2) \\ -\frac{1}{8}b(s-1)(t+1)^2(t-1) \\ -\frac{1}{8}a(s-1)^2(s+1)(t+1) \end{array} \right\}$$

3

单元构造：构造单元-矩形板

- 组装刚度阵流程
 - 建立当地坐标系
 - 利用给定形函数得到当地矩阵
 - 转换到全局坐标系

$$K_1 =$$

$$\begin{bmatrix} 0 & 20 & & & \\ 0 & 0 & 60p^{-2} & & \\ 0 & 30p^{-1} & 0 & 0 & \text{Symmetrical } p = \frac{a}{b} \\ 0 & 0 & 0 & 0 & \\ 0 & 10 & 15p^{-1} & 0 & 20 \\ 0 & 15p^{-1} & 30p^{-2} & 0 & 30p^{-1} & 60p^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 30p^{-1} & 0 & 5 & 15p^{-1} & 0 & 20 \\ 0 & -30p^{-1} & -60p^{-2} & 0 & -15p^{-1} & -30p^{-2} & 0 & -30p^{-1} & 60p^{-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 15p^{-1} & 0 & 10 & 30p^{-1} & 0 & 10 & -15p^{-1} & 0 & 20 \\ 0 & -15p^{-1} & -30p^{-2} & 0 & -30p^{-1} & -60p^{-2} & 0 & -15p^{-1} & 30p^{-2} & 0 & -30p^{-1} & 60p^{-2} \end{bmatrix}$$

$$K_2 =$$

$$\begin{bmatrix} 20 & & & & \\ 0 & 0 & & & \\ -30p & 0 & 60p^2 & & \text{Symmetrical} \\ 10 & 0 & -30p & 20 & \\ 0 & 0 & 0 & 0 & 0 \\ 30p & 0 & -60p^2 & 30p & 0 & 60p^2 \\ 10 & 0 & -15p & 5 & 0 & 15p & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -15p & 0 & 30p^2 & -15p & 0 & -30p^2 & -30p & 0 & 60p^2 \\ 5 & 0 & -15p & 10 & 0 & 15p & 10 & 0 & -30p & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15p & 0 & -30p^2 & 15p & 0 & 30p^2 & 30p & 0 & -60p^2 & 30p & 0 & 60p^2 \end{bmatrix}$$

$$K_3 =$$

$$\begin{bmatrix} 0 & 0 & & & \\ -15p^{-1} & 0 & 15p & 30 & \\ 0 & 0 & 0 & 0 & \text{Symmetrical} \\ 0 & 0 & -15p & 15 & 0 \\ 0 & -15p & -30 & 15p^{-1} & 15p & 30 \\ 0 & 0 & 15p^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 \\ 15p^{-1} & 0 & -30 & 0 & 0 & 30 & -15p^{-1} & -15p & 30 \\ 0 & 0 & 0 & 0 & 0 & -15p^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15p & -15 \\ 0 & 0 & 30 & -15p^{-1} & 0 & -30 & 0 & 15p & -30 & 0 & 15p^{-1} & -15p & 30 \end{bmatrix}$$

3

单元构造：构造单元-矩形板

- 应力输出
 - 向 TECPLOT 输出四节点的应力
 - 通过外推法得到，板单元的二次高斯点是超收敛点
- 自由度设置
 - 对于输入时旋转自由度不作处理
 - 对于法向沿坐标轴的板，仅放开另外两个方向的自由度，其他方向则开放三个方向的自由度

3

单元构造：构造单元-平板壳

- 单元实现: $4Q$ +板单元
- 和板单元主要的不同
 - 刚度阵增加 $4Q$ 项
 - 增加 $4Q$ 对应应力

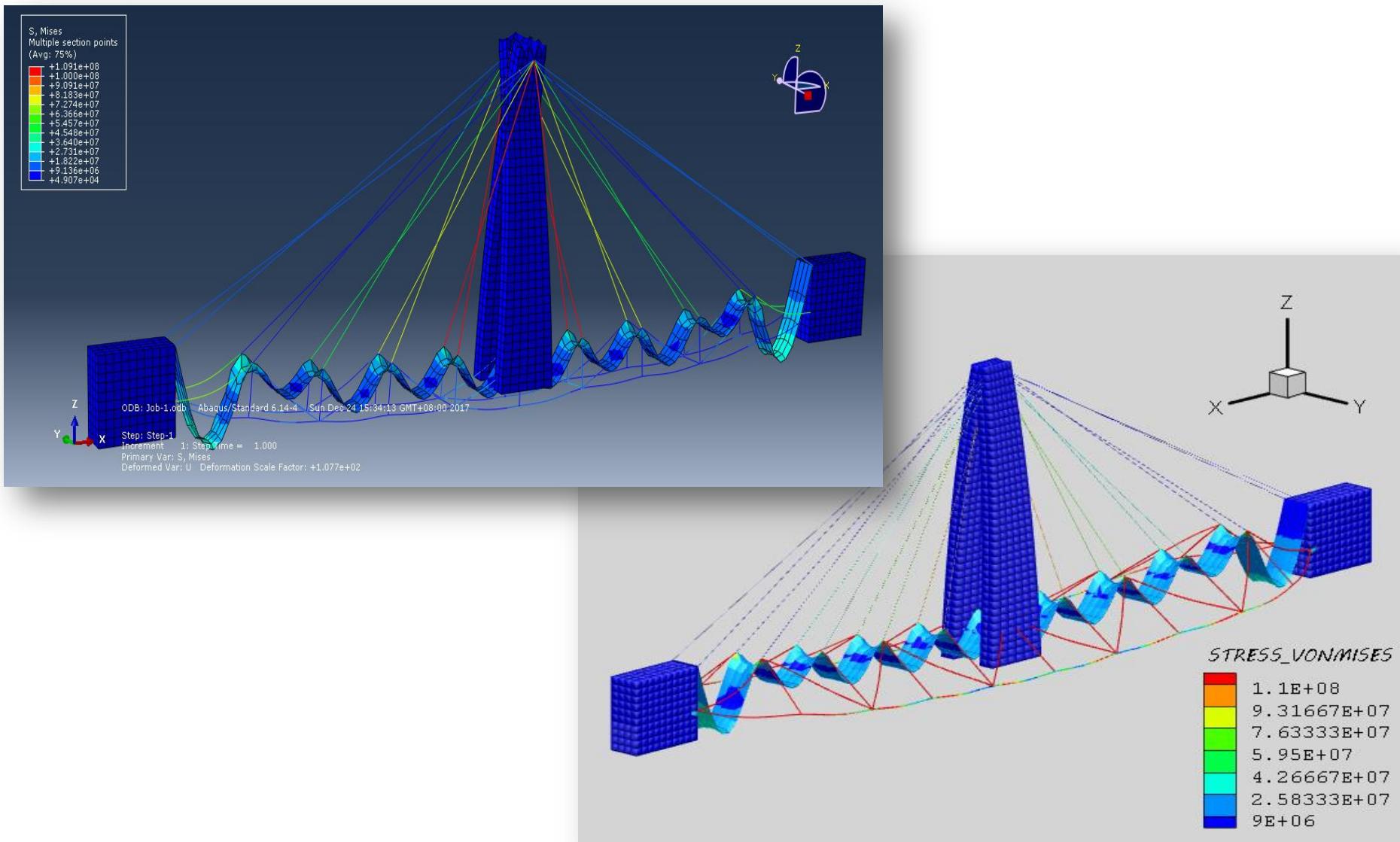


高级功能

模态分析
前处理与稀疏求解器
后处理与SPR

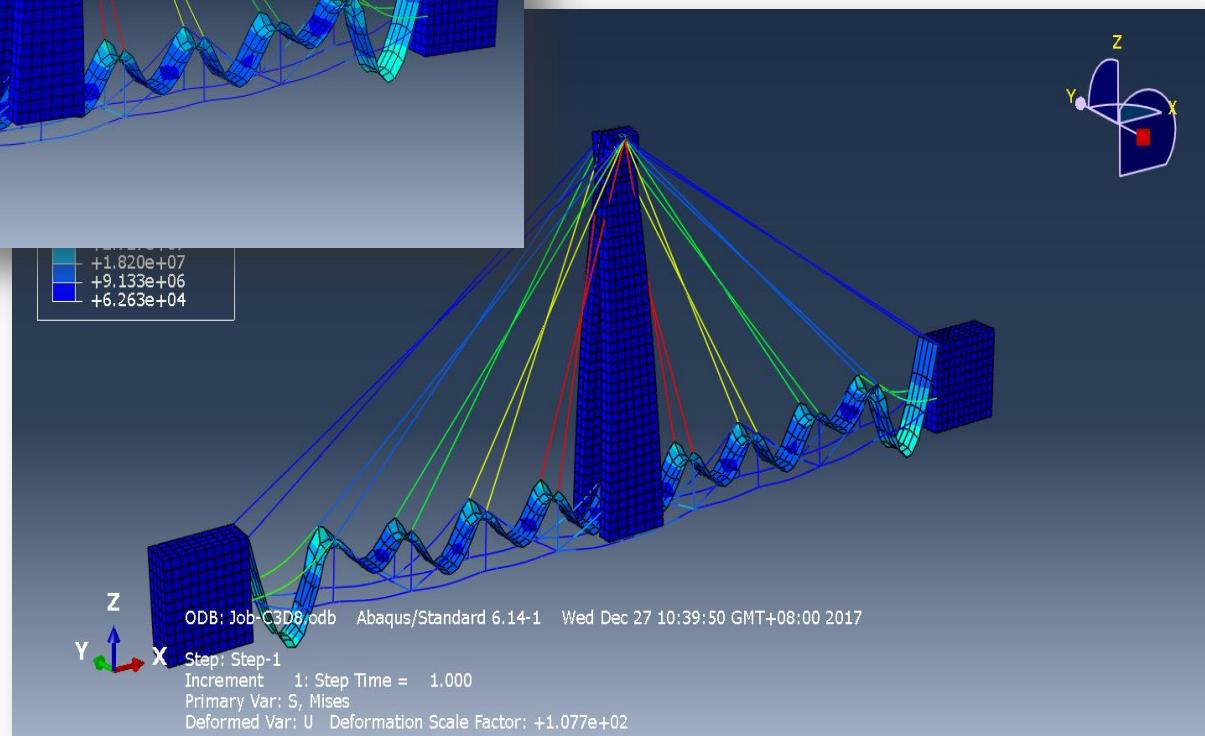
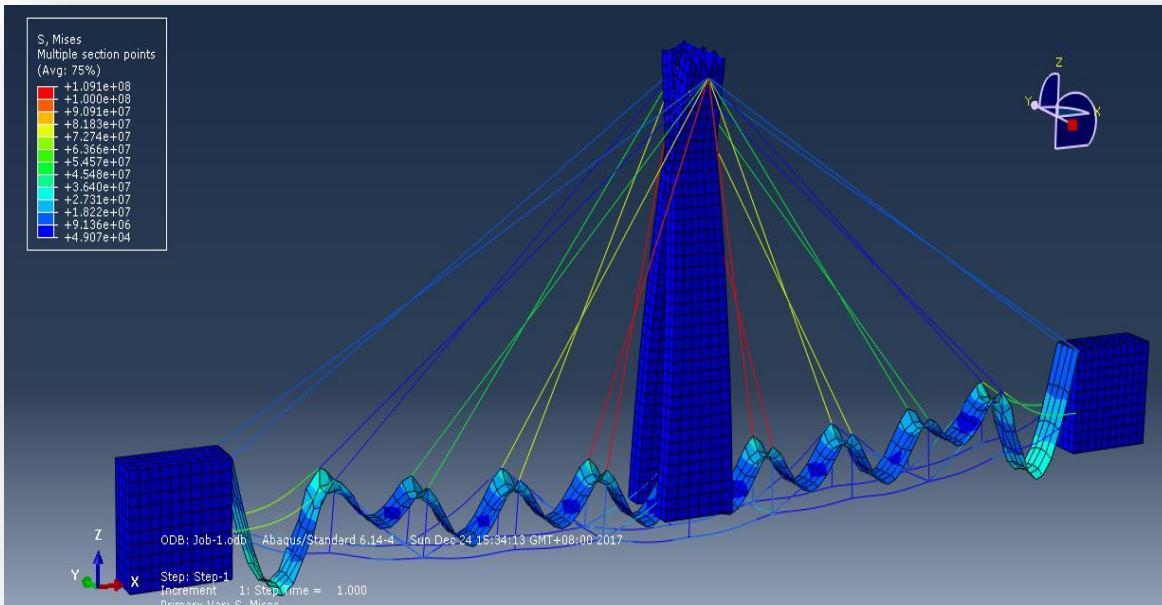


结果对比





结果对比



4

高级功能：模态分析

一、基本思路

- 使用已有刚度阵
- 组装一致质量矩阵（目前只有 Bar 单元）
- 解广义特征值问题（子空间迭代法）
 - 缩减前的部分利用 **LDLTSolver** 计算
 - 缩减后的部分利用 **MKL** 库的解广义特征值问题的函数

4

高级功能：模态分析

二、算例验证：一维质弹系统



- 对于完整的阶数能够给出精确解
- 对于求前两阶特征值的情况：
 - 第一阶特征值基本准确
 - 第二阶特征值稍有误差

4

高级功能：前处理

一. 内存优化

- 对于前处理，跑完节点编号部分后，计算
- 在计算完节点编号之后输出临时文件，清

二. 输入文件格式优化

- 现有 **dat** 格式输入能力有局限
 - 纯文本需要手动解析，需要提前写好
- 提出一个新文件输入的格式：**json** 格式
 - 流行，有高效现成解析库可以调用
 - 相对 **xml** 省空间
 - 可读性好、可扩展性好，不管是手写还是与其他程序交换都简单

```
1  {
2      "description": "heading line",
3      "analyse type": "static",
4      "node info": [
5          [1, 0, 0, 0, 1.0, 1.0, 1.0],
6          [2, 0, 0, 0, 1, 1, 1, 2.0, 2.0]
7      ],
8      "load data": [
9          [
10             {
11                 "type": "node force",
12                 "forces": [
13                     [1, 1, 1.0]
14                 ]
15             },
16             {
17                 "type": "body force",
18                 "force": [3, 9.8]
19             }
20         ]
21     ],
22     "elements": [
23         {
24             "element type": "bar",
25             "materials": [
26                 [1e6, 3]
27             ],
28             "elements": [
29                 [1, 2, 1]
30             ]
31         }
32     ],
33     "interactions": []
34 }
35 }
```

4

高级功能：稀疏求解器

添加稀疏求解器

1. 矩阵类替换：用 CSR 矩阵替代 Skyline 矩阵
 - 用 `std::set` 或 `std::vector` 来实现储存 `columns`, 同时计算 `rowIndex`
 - 用 `STL` 的好处：
 - 使用灵活快速，不用重复造轮子
 - `STL` 容器有 `Allocator` 避免内存碎片化
 - `trim` 负优化
2. 求解器替换：用 MKL PARDISO 求解器替代 LDLT 求解器
 - 多线程，用 `MKL_NUM_THREADS` 控制线程数量
 - 自动切换 IC/OOC 模式，控制内存大小和缓存路径

4

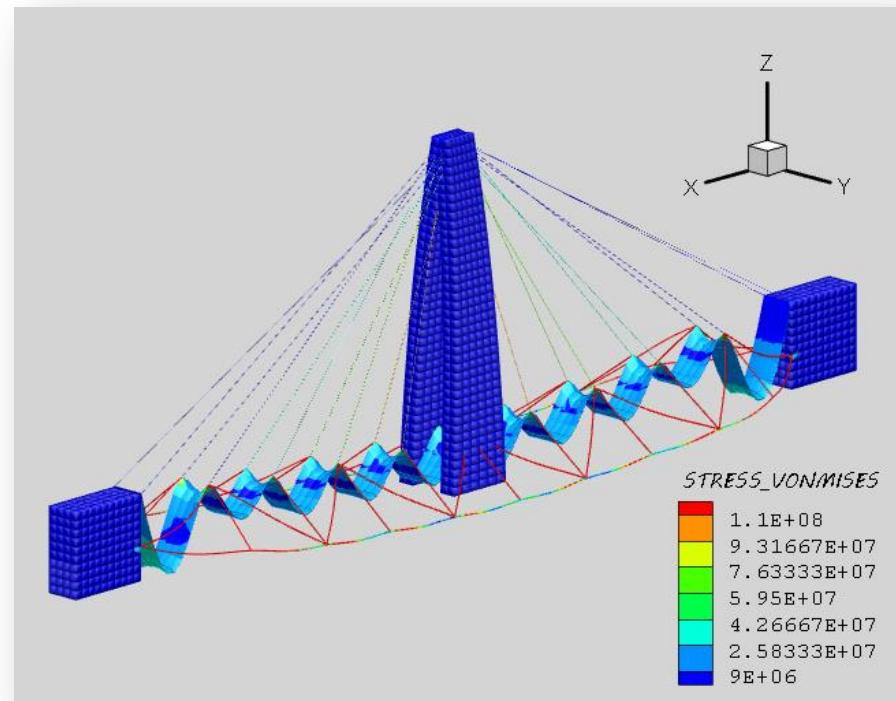
高级功能：后处理

一、后处理环境的选择

- Tecplot
- Paraview

二、软件使用探究

- 杆单元绘制
- 单元节点输入顺序
- 使用 ZONETYPE 和使用 PEPOINT 的区别
- 输出效果调试



4

高级功能：后处理

三、算法实现

第一部分：读入输入文件名filename及控制信息，写入“filename_post.dat”

第二部分：读入单元组（单元类型）信息，该单元组所包含的单元数目、对应的TECPLOT画图类型等。

第三部分：加入放大因子coeff，按序输出结点信息：

1. 形变后坐标： $x_{post}, y_{post}, z_{post}$
2. 三个应力不变量： $\sigma_1, \sigma_2, \sigma_3, \sigma_{vonMises}$
3. 六个应力分量： $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \tau_{xy}, \tau_{yz}, \tau_{zx}$

按顺序输出每个单元的结点编号（这里的编号顺序按上方位移的输出顺序）

4

高级功能：后处理

四、后处理优化

- 应力平滑（单元内部）
- SPR（单元组内部）
- 后处理分量

4

高级功能：基于 8H 的 SPR

一、程序原理

1. 三维完备二阶多项式，共有 10 项；
2. 内部点、边点、角点分开讨论
 - 边点、内部点：对于内部点和边点，取它所属于的所有单元作为一个分片；
 - 角点：所在分片的算术平均值。

4

高级功能：基于 8H 的 SPR

二、算法实现



Finite Elements in Analysis and Design

Volume 19, Issues 1–2, February 1995, Pages 11-23



Superconvergence and the superconvergent patch recovery

O.C. Zienkiewicz ^a, J.Z. Zhu ^b

- 1) 按照单元组循环，结点分类
- 2) 处理边点和内点，构造分片
- 3) 处理角点，进行算术平均
- 4) 按照 TECPLOT 格式输出结点坐标和对应应力值



STAP90 v.s. STAPpp

PROS

1. OOC（面向对象）模式实现，自然易懂
2. 用单例设计模式避免全局变量，保证内存有效使用
3. 不同模块间解耦，方便合作
4. 宇宙最强 IDE —— VS 的免费支持，调试方便快速
5. 网络上大量关于 C++ 的资料

CONS

1. 第一年，框架尚不成熟
2. C++ 基础欠缺，课程体系缺少 C++ 课程
3. 倾向于 C with classes，应该用更多 C++ 的风格 (template, call-by-reference, polymorphism)

5

致谢

- 感谢一直以来组内每一位同学的不懈努力以及团队所有成员的通力合作！
- 感谢张老师和宋言学长在整个项目完成工作中对本组全组同学的耐心指导！
- 感谢在部分任务中，其余各组的部分同学与本组同学的积极讨论与互相扶持！



参考文献

1. 张雄 等. 计算动力学 (第 2 版). 北京: 清华大学出版社, 2015.
2. 彭细荣 等. 有限单元法及其应用. 北京: 清华大学出版社, 北京交通大学出版社. 2012
3. Zienkiewicz O C, Cheung Y K. The Finite Element Method for Analysis of Elastic Isotropic and Orthotropic Slabs. Proc. Inst. Civ. Eng. 28:471-488, 1964.
4. <https://software.intel.com/en-us/mkl-developer-reference-c-spgvd#22ACC038-0667-4E23-8263-B4C0E67A769C>
5. 陆明万. 弹性理论基础 (第2版. ed). 北京, 德国: 清华大学出版社, 施普林格出版社, 2001.



参考文献(续)

6. 青克维奇, O.C., 泰勒, R.L. & 曾攀 x. 有限元方法 (第1卷, 基本原理, volume 1 the basis), 北京: 清华大学出版社, 2008 .
7. 王勣成 . 有限单元法 . 北京 : 清华大学出版社, 2003 .
8. 薛守义 . 有限单元法 . 北京 : 中国建材工业出版社, 2005 .
9. Wikipedia. Timoshenko beam theory. (2017.08.25) [2017.12.06].
https://en.wikipedia.org/wiki/Timoshenko_beam_theory.



THANKS